

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 7

### REMARKS

#### Claims 1-8, 10-14, and 16-19

The Examiner has indicated that claims 1-8, 10-14, and 16-19 have been rejected under 35 USC 103(a) as being unpatentable over EFI Application Toolkit in view of Goodman (2002/0091807). However, the Examiner also relies upon Noel (2002/0010844), and modifies the prior art of EFI in view of Goodman per Noel. Therefore, Applicant notes that this rejection is more properly made such that the pending claims are rejected under 35 USC 103(a) as being unpatentable over EFI in view of Goodman, and further in view of Noel. If the Examiner believes that Applicant is incorrect, he is requested to indicate how Noel is being used if not to show limitations of the claimed invention. Otherwise, he is requested to properly describe the rejection that he is proffering. Applicant responds to this rejection as if it were made over EFI in view of Goodman and further in view of Noel, since this is the substance of the Examiner's rejection.

Claims 1, 10, and 18 are independent claims, from which the remaining rejected claims ultimately depend. Applicant submits that, at least as amended, claims 1, 10, and 18 are patentable over EFI in view of Goodman and further in view of Noel. Therefore, for at least the same reasons, all the pending claims are patentable over EFI in view of Goodman and further in view of Noel.

Claim 1 is discussed herein as representative of all the independent claims 1, 10, and 18, insofar as patentability over EFI in view of Goodman and further in view of Noel is concerned. Claim 1 has been amended as follows. First, each of the nodes has been amended to include "one or more processors, memory, and input/output (I/O) modules." Thus, a node is not just a processor, but includes processor(s), memory, and I/O modules. Support for this amendment is found in the patent application as filed at least in page 6, paragraph 28, and page 1, paragraph 6.

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 8

Second, the process that the base node instructs the other nodes to perform has been amended to relate to “updating the firmware” of the other nodes. As such, the firmware of the other nodes “is updated without rebooting of any” of the other nodes. Support for this amendment is found in the patent application as filed at least in page 8, paragraph 34, and page 11, paragraph 43.

It is instructive now to consider what the claimed invention is directed to. A base node instructs at least one another node to start a process related to firmware of the other node, where each node includes processors, memory, and I/O modules, and the process relates to updating the firmware of the other nodes. The process is started by each other node, and each other node reports back the results of the process to the base node. The method is performed without rebooting of any node, such that the firmware of each other node is updated without rebooting of these nodes.

Applicant submits that the claimed invention at least as has been amended is patentable over EFI in view of Goodman and further in view of Noel, for at least FIVE *independent and separate* reasons. First, EFI in view of Goodman and further in view of Noel does not teach each node include including processors, memory and I/O modules, at least insofar as reliance upon EFI as the primary and base reference is concerned. Second, EFI is not properly combined with Goodman or Noel, since EFI has a different definition of a node as compared to Goodman and Noel. Third, EFI in view of Goodman and further in view of Noel does not have a base node instructs at least one other node to start a process relating to updating the firmware of the other node. Fourth, EFI in view of Goodman and further in view of Noel does not update firmware of a node without rebooting the node. Fifth, there is no stated motivation, and indeed there is contrary motivation, to modify EFI in view of Goodman and further in view of Noel to update the firmware of a node without rebooting the node. It is noted that each of these reasons for patentability is independent and separate. That is, any of them is a reason why the claimed invention is patentable.

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 9

Applicant respectfully requests that the Examiner consider ALL of these independent reasons for patentability, and if he disagrees with Applicant, that he discuss in his next office action ALL of the independent reasons for patentability. Discussing just one or two of these independent reasons for patentability is not a complete consideration nor a complete response to Applicant.

*First independent reason of patentability*

First, EFI in view of Goodman and further in view of Noel does not teach each node include including processors, memory and I/O modules, at least insofar as reliance upon EFI as the primary and base reference is concerned. The Examiner relies upon EFI in particular as teaching the claimed base node and the claimed at least one other node. In particular, the Examiner states that BSP and AP of FIG. 1 of EFI are the base node and the at least one other node, respectively, of the claimed invention.

However, BSP and AP of EFI are processors, a "boot processor" and a "non-boot processor" respectively. (EFI, p. 3-3) Both of these processors are components of a multiple-processor protocol (MPP) architecture, in "a shared memory environment." (Id.) Applicant asserts, then, that one of ordinary skill within the art would most likely interpret EFI such that the processors are *of the same node*, and not *of different nodes*, as is required by the claimed invention.

The reasoning for this assertion is as follows. First, EFI only makes reference to a shared memory environment. That is, the processors of EFI share the same memory. This is comparable to the node 302 of FIG. 3 of the patent application as filed, in which processors 308 have shared memory 304. At best, this is what EFI discloses. EFI discloses a multiple-processor system, not necessarily a multiple-node system, which is inherently more complex.

Stated another way, there is no discussion within EFI of separate nodes encompassing the processors BSP and AP, where such separate nodes are interconnected over a network. For

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 10

instance, compare EFI to FIG. 2 of the patent application as filed, in which there are a number of nodes 202 and 204 communicating over a network 206. If EFI intended for its processors to be part of separate nodes, it stands to reason that it would have disclosed this fact, and show how its nodes can communicate with one another, such as over a network.

Therefore, EFI fails as a base reference, such that EFI in view of Goodman and further in view of Noel does not disclose all the claimed limitations of the invention. It is not apparent, for instance, that the communication protocol of EFI is workable where the processors are part of separate nodes. Issues such as latency, network congestion, and so on, can affect the communication protocol described in EFI from working where the processors are part of separate nodes. If the Examiner wishes to continue relying on EFI, he is requested by Applicant to show a reference in which the EFI protocol is depicted as being workable over a multiple-node system, as opposed to simply over a multiple-processor system.

*Second independent reason of patentability*

Second, EFI is not properly combined with Goodman or Noel, since EFI has a different definition of a node as compared to Goodman and Noel. Stated another way, the Examiner has to show that EFI is properly extended to a multiple-node system, as opposed to simply a multiple-processor system as explicitly disclosed in EFI. As has been described above, EFI shows multiple processors, which are not necessarily part of different, multiple nodes.

By comparison, both Goodman and Noel show multiple nodes. For instance, FIG. 1 of Goodman shows nodes 20, 40, 60, and 80, each of which contains its own processors 22. Similarly, FIG. 1 of Noel shows nodes 102, 104, and 106, each of contains its own processors 108, 110, 112, and 14. By comparison, EFI has been construed by the Examiner as having one of its processors BSP be one node, and another of its processors AP be another node.

Therefore, EFI is on its face incompatible with Goodman and Noel. EFI is discussing individual processors, whereas Goodman and Noel are discussing individual nodes. EFI makes no

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 11

mention of mapping its processors to separate nodes. Indeed, insofar as it talks about a multiple-processor system, as opposed to a multiple-node system, the most likely interpretation to one of ordinary skill within the art is that the processors of EFI are part of the same node or system. The only motivation to consider using EFI's multiple processors as separate nodes comes from the claimed invention itself. That is, the only motivation to combine EFI with Goodman and Noel to map EFI's processors as part of the separate nodes of Goodman and Noel comes from the claims.

If the Examiner wishes to continue to rely on EFI, he is respectfully requested to provide a reference indicating that the EFI protocol can be employed within a multiple-node system, instead of simply within a multiple-processor system. Without such an explicit reference, the Examiner cannot use the claimed invention to inform him to interpret EFI to a multiple-node system. Doing so would use impermissible hindsight. Therefore, as it stands now, EFI is incompatible with and thus cannot be combined with Goodman and Noel.

*Third independent reason of patentability*

Third, EFI in view of Goodman and further in view of Noel does not have a base node instructs at least one other node to start a process relating to updating the firmware of the other node. The Examiner has construed EFI's StartProcessor() function on page 4-11 as corresponding to the base node instructing at least one other node to start a process at the other node. However, Applicant notes that the StartProcessor() function of EFI has nothing to do with updating the firmware of the other node. Indeed, insofar as EFI discusses processors, and not nodes, the StartProcessor() function has nothing to do with firmware, since nodes have firmware, not individual processors.

*Fourth independent reason of patentability*

Fourth, EFI in view of Goodman and further in view of Noel does not update firmware of a node without rebooting the node. Of all three of these references, only Goodman discusses

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 12

updating the firmware of a node. However, Goodman specifically reboots a node after its firmware has been updated. As stated in paragraph [0029] and in FIG. 4 of Goodman, the memory of a node is updated “to include the new version of the firmware.” Then the nodes are “reset through a remote reset or reboot message, electrically signal, or manual process, e.g., switching the power on and off, etc.” Thus, Goodman, and therefore EFI in view of Goodman and further in view of Noel, results in updating the firmware *with* rebooting the nodes, and *without* rebooting the nodes, as in the claimed invention.

The Examiner has contended that Noel shows performing a process related to the firmware of a node without having to reboot the node. First, however, the claimed invention is now limited to updating the firmware of a node without having to reboot the node, which Noel does not disclose. Second, Applicant very respectfully asserts that the Examiner is very much overreaching in stating that Noel shows rebooting not being necessary.

The Examiner states that “memory allocation of fragments” can be changed by modifying “the configuration tree” without the need for a reboot in paragraph [0286] of Noel. However, it is difficult to see how one of ordinary skill within the art would interpret “changing memory allocation of fragments by modifying the configuration tree” as updating the firmware of a node. Updating the firmware of a node is a very common practice within the prior art, and if Noel indeed were performing this sort of functionality, then it would have explicitly said so.

The Examiner tries to tie in this statement to the firmware of the node by stating that the configuration tree is stored in local address space in paragraph [0070] of Noel, where the tree is altered. However, paragraph [0070] of Noel really states that “an operating system instance *maps* the tree into *its* local address space.” The tree is not necessarily stored in local address space, it is merely *mapped* to the operating system’s local address space, so that the tree is easier to modify – regardless where the tree is stored.

More significantly, however, local address space does not necessarily mean local memory, indeed local memory does not necessarily mean private memory, and private memory does not

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 13

necessarily mean firmware. First, a local address space can have mapped thereto data that is stored elsewhere, so that a local address space is not the same as local memory. For example, a data structure stored on remote memory can nevertheless be mapped to a local address space, so that the structure can be modified – as stored in remote memory – by using the local address space. Therefore, the Examiner to rely upon Noel as he does, he first has to show that Noel specifically is using the terms local address space and local memory as synonymous. However, Noel is not using these terms synonymously, insofar as the operating system *maps* a tree (stored elsewhere) into local address space.

Second, local memory does not necessarily mean private memory. Consider, for instance, a non-uniform memory architecture (NUMA), in which each node has local memory that is shared by the other nodes as remote memory to the other nodes. In this case, local memory is not private to a node, but is shared by the other nodes. Therefore, Applicant respectfully requests the Examiner to show where in Noel the local memory of a node is indicated specifically to be private memory and not shared memory – and furthermore that the local memory where a tree is stored is indeed private memory and not shared memory.

Third, private memory does not mean firmware. There are many types of private memory, and firmware is only one type. If Noel really intended for its private memory to be firmware, then it would have said firmware. Indeed, it stands to reason that Noel is not talking about firmware, insofar as its nodes are partitioned into partitions, which does not necessarily have anything to do with firmware. Therefore, Applicant respectfully requests the Examiner to show where in Noel the private memory of Noel *is used to store a tree*, such that the private memory is actually firmware.

Indeed, Noel notes in paragraph [0286] that some changes require rebooting and other changes do not require rebooting. Moving an I/O processor from one partition to another, for instance, is indicated as requiring rebooting. Updating the firmware is not mentioned in Noel at all. Therefore, Noel cannot be relied upon as stating that firmware updating does not require

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 14

rebooting. If it is, the only way to guide interpretation of Noel such that modifying a tree is comparable to updating firmware, but moving an I/O processor from one partition to another is not comparable to updating firmware, where the former does not result in rebooting in Noel and the latter does result in rebooting in Noel, is to employ impermissible hindsight, by using the claimed invention as a template to patch together the prior art.

In summary, the Examiner makes too many jumps in logic in making this rejection on this basis. The claimed invention is concerned with updating the firmware of a node without having to reboot the node. The prior art just does not show how firmware of a node can be updated without having to reboot the node. Rather, the only prior art that specifically discusses firmware updating is Goodman – and Goodman specifically says you have to reboot the node when up date firmware.

*Fifth separate reason for patentability*

Fifth, there is no stated motivation, and indeed there is contrary motivation, to modify EFI in view of Goodman and further in view of Noel to update the firmware of a node without rebooting the node. Let us look at what these references teach, as has been described in detail above. EFI does not say anything about updating the firmware. Goodman says that to update firmware, you have to reboot a node. Noel also does not say anything about updating firmware, but says that some changes may require rebooting of a node, and that other changes may not require rebooting a node.

Therefore, if you combine all three of these references, there is no motivation to modify EFI in view of Goodman and further in view of Noel to update firmware without rebooting a node. Goodman actually says the opposite, such that combining it with EFI teaches away from the claimed invention. If you add Noel into the mix, combining it with EFI in view of Goodman, the only way to make Noel consistent with EFI in view of Goodman is to say that updating firmware is not something that can be done without rebooting.



McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

---

Page 15

That is, if you want to read Noel as saying that firmware can be updated without rebooting, you are only doing this by relying upon impermissible hindsight. This is why. First, Noel does not actually mention anything about updating firmware. Second, Noel discloses that some changes require rebooting, and some changes do not require rebooting. Therefore, in which camp do you put updating firmware? If you go by the prior art – specifically Goodman – then updating firmware is a change that requires rebooting. If you go by the claimed invention, then updating firmware is a change that does not require rebooting.

However, obviousness analysis requires that you do not use impermissible hindsight – that you do not use the claimed invention as a template to piece together the prior art. Therefore, if you look only to the prior art, then you have per Goodman the updating of firmware as requiring rebooting. The only way you have updating of firmware without requiring rebooting is by looking to the prior art, which is impermissible.

#### *Summary*

Applicant has provided five separate and independent reasons why the claimed invention is patentable over EFI in view of Goodman, and further in view of Noel. For the rejection to stand, the Examiner has to specifically refute all five of these reasons. Applicant requests that the Examiner therefore discuss *each reason* in his final office action, should the claimed invention not be allowed.

McAdams  
Serial no. 09/921,841  
Filed 8/2/2001  
Attorney docket no. BEA920010012US1

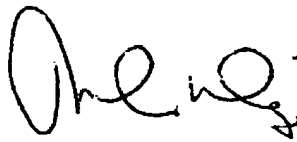
---

Page 16

Conclusion

Applicants have made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Applicants' Attorney so that such issues may be resolved as expeditiously as possible. For these reasons, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,



May 23, 2006  
Date

Michael A. Dryja, Reg. No. 39,662  
Attorney/Agent for Applicant(s)

Law Offices of Michael Dryja  
704 228<sup>th</sup> Ave NE #694  
Sammamish, WA 98074  
tel: 425-427-5094  
fax: 206-374-2819